

## Laboratorio Nro. 5 Implementación de Grafos

### Objetivos

1. Entender la implementación de los grafos dirigidos
2. Entender el concepto de sucesor (*successor*) como vecino, es decir, un nodo que es vecino de otro, un nodo que está conectado a otro por un arco.

### Consideraciones Iniciales

#### Leer la Guía



Antes de comenzar a resolver el presente laboratorio, leer la **“Guía Metodológica para la realización y entrega de laboratorios de Estructura de Datos y Algoritmos”** que les orientará sobre los requisitos de entrega para este y todos los laboratorios, las rúbricas de calificación, el desarrollo de procedimientos, entre otros aspectos importantes.

#### Registrar Reclamos



En caso de tener **algún comentario** sobre la nota recibida en este u otro laboratorio, pueden **enviarlo** a través de <http://bit.ly/2q4TTKf>, el cual será atendido en la menor brevedad posible.

**Traducción de Ejercicios**

En el GitHub del docente, encontrarán la traducción al español de los enunciados de los Ejercicios en Línea.

**Visualización de  
Calificaciones**

A través de **Eafit Interactiva** encontrarán **un enlace** que les permitirá **ver un registro de las calificaciones** que **emite el docente** para cada taller de laboratorio y según las rubricas expuestas. **Véase sección 3, numeral 3.7.**

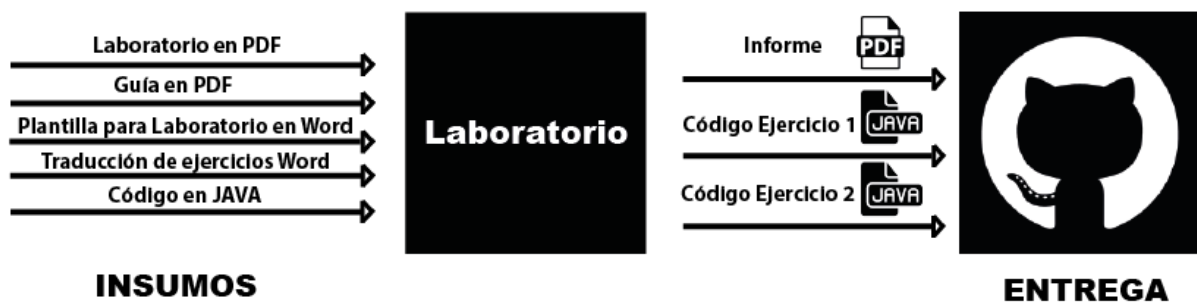
**GitHub**

Crear un repositorio en su cuenta de GitHub con el nombre `st0245-suCodigoAqui`. **2.** Crear una carpeta dentro de ese repositorio con el nombre `laboratorios`. **3.** Dentro de la carpeta `laboratorios`, crear una carpeta con nombre `lab06`. **4.** Dentro de la carpeta `lab06`, crear tres carpetas: `informe`, `codigo` y `ejercicioEnLinea`. **5.** Subir el informe pdf a la carpeta `infome`, el código del ejercicio 1 a la carpeta `codigo` y el código del ejercicio en línea a la carpeta `ejercicioEnLinea`. Así:

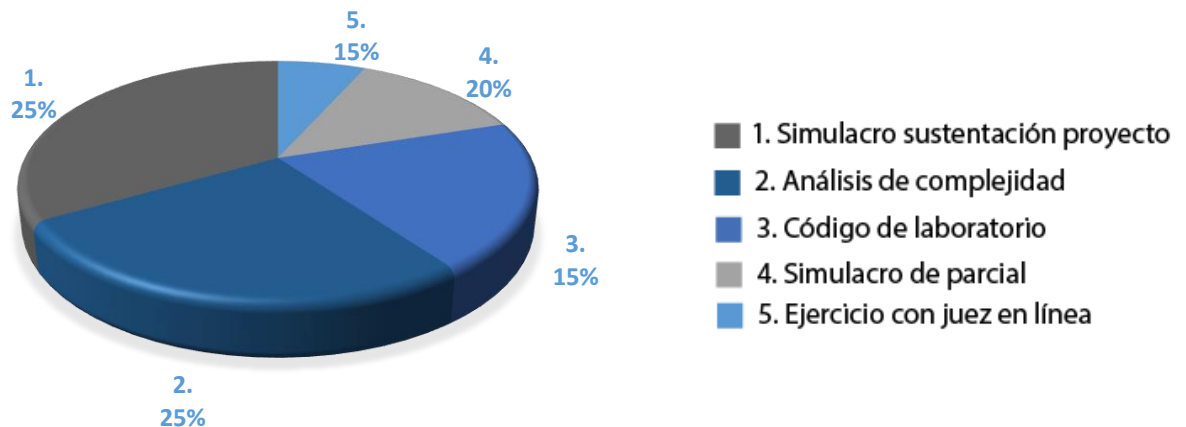
```
st0245-suCodigoAqui
  laboratorios
    lab01
      informe
      codigo
      ejercicioEnLinea
    lab02
    ...
```

## Intercambio de archivos

Los archivos que **ustedes deben entregar** al docente son: **un archivo PDF** con el informe de laboratorio usando la plantilla definida, y **dos códigos**, uno con la solución al numeral 1 y otro al numeral 2 del presente. Todo lo anterior se entrega en **GitHub**.



## Porcentajes y criterios de evaluación para el laboratorio



## Resolver Ejercicios

### 1. Códigos para entregar en GitHub:



En la vida real, la documentación del software hace parte de muchos estándares de calidad como CMMI e ISO/IEC 9126



Véase Guía **en Sección 3, numeral 3.4**



Código de laboratorio en **GitHub**. Véase Guía en **Sección 4, numeral 4.24**



*Entregar documentación en **JavaDoc** o equivalente. El uso de JavaDoc es opcional*



**No se reciben** archivos en **.RAR** ni en **.ZIP**



En la vida real, los grafos se utilizan para representar redes sociales como *Facebook*, sistemas de información geográfica como *Google Earth* o enrutadores, como un enrutador ISR 4000 de Cisco

#### 1.1 Teniendo en cuenta lo anterior:

- a) Realicen una implementación de la clase abstracta *Digraph*, llámela *DigraphAM* e implementen grafos con la estructura de datos Matrices de Adyacencia Etiquetadas
- b) Posteriormente, creen la clase *DigraphAL* e implementen grafos con la estructura de datos Listas de Adyacencia. Ambas clases heredan de la clase abstracta *Digraph* (digrafo o grafo dirigido)

**1.2** En la clase *GraphAlgorithms*, implementen un método que reciba como parámetro un grafo dirigido y que retorne cuál es el vértice que tiene más sucesores (vecinos). Debe funcionar para ambas implementaciones de grafo.







**1.3** Prueben su código con los ejemplos contruidos en los numerales 1.1 y 1.2 para el *Algoritmo de Dijkstra*. Deben obtener la misma respuesta con ambas implementaciones.



En la vida real, una aplicación de los grafos es para describir mapas, como los usados por Google Maps. El archivo *medellin\_colombia-grande.txt* que está en el ZIP que el docente les entregó, contiene un grafo que representa todas las calles de Medellín, es un grafo de aproximadamente 300.000 nodos

**1.4** Teniendo en cuenta lo anterior, implementen un método que permita crear un grafo a partir de ese archivo del texto *medellin\_colombia-grande.txt*

## 2) Ejercicios en línea sin documentación en GitHub:

	Véase Guía en <b>Sección 3, numeral 3.3</b>		<b>No entregar</b> documentación <b>HTML</b>
	<i>Entregar un archivo en <b>.JAVA</b></i>		<b>No se reciben</b> archivos en <b>.PDF</b>
	<b>Resolver</b> los problemas de <b>CodingBat</b> usando <b>Recursión</b>		Código del ejercicio en línea en <b>GitHub</b> . Véase Guía en <b>Sección 4, numeral 4.24</b>

### 2.1 Resuelvan el siguiente ejercicio

En 1976, el teorema de colorear un mapa con 4 colores fue probado con la ayuda de un computador. Este teorema muestra que un mapa puede ser coloreado solamente con 4 colores, de tal forma que no haya una región coloreada usando el mismo color que un vecino. Aquí hay un problema similar a ese problema, pero es mucho más simple.

Usted tiene que decidir si dado un grafo conexo arbitrario, ese grafo se puede colorear con 2 colores. Esto quiere decir, si uno puede asignar colores (de una paleta de 2 colores) a los nodos, de tal forma que no haya 2 nodos adyacentes del mismo color. Para simplificar el problema usted puede asumir que:

1. No hay un nodo que tenga un arco a sí mismo
2. El grafo es no dirigido, es decir que, si un nodo *a* está conectado a un nodo *b*, usted puede asumir que el nodo *b* también está conectado al nodo *a*.

3. El grafo será fuertemente conexo. Esto quiere decir, que hay al menos un camino de un nodo de grafo a cualquier otro nodo.

### Entrada

La entrada consiste en varios casos de prueba. Cada caso de prueba comienza con una línea que tiene un número  $n$  ( $1 < n < 200$ ) de nodos diferentes. La siguiente línea contiene el número de arcos.

Posteriormente, las siguientes líneas, cada una contiene 2 número que especifican que existe un arco entre dos nodos.

Un nodo en el grafo se representa con un número  $a$  ( $0 < a < n$ ). Una entrada con  $n = 0$  simboliza el fin de la entrada y no debe ser procesada.

### Salida

Usted tiene que decidir si el grafo de entrada puede ser coloreado con dos colores o no, y debe imprimirlo como se muestra a continuación.

### Entrada de los ejemplos

```
3
3
0 1
1 2
2 0
3
2
0 1
1 2
9
8
0 1
0 2
0 3
0 4
0 5
```

0 6  
0 7  
0 8  
0

### Salida de los ejemplos

NOT BICOLORABLE.  
BICOLORABLE.  
BICOLORABLE.



**Entregar** un archivo  
en **.JAVA**



O **entregar** un archivo  
en **.CPP**



O **entregar** un  
archivo en **.PY**

**2.2[Ejercicio Opcional]:** Resuelvan el siguiente problema <http://bit.ly/2gTLZ53>



Pueden **entregar** un  
archivo en **JAVA**



O **entregar** un archivo  
en **.CPP**



O **entregar** un  
archivo en **.PY**



### 3) Simulacro de preguntas de sustentación de Proyectos



Véase Guía en **Sección 3,**  
**Numeral 3.4**



Entregar informe de  
laboratorio en **PDF**



Usen la **plantilla** para  
responder laboratorios



**No apliquen Normas**  
**Icontec** para esto

**3.1** Incluyan una imagen de la respuesta de las pruebas del numeral 1.3

**3.2** Escriban una explicación entre 3 y 6 líneas de texto del código del numeral 1.1. Digan cómo funciona, cómo está implementado el grafo con matrices y con listas que hizo, destacando las estructuras de datos y algoritmos usados

**3.3** ¿En qué grafos es más conveniente utilizar la implementación con matrices de adyacencia y en qué casos es más convenientes listas de adyacencia? ¿Por qué?

**3.4** Para representar el mapa de la ciudad de Medellín del ejercicio del numeral 1.3, ¿qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?



En la vida real para una red social como **Facebook**, donde hay al menos 100 millones de usuarios, pero cada usuario tiene en promedio 200 amigos,

**3.5** Teniendo en cuenta lo anterior, respondan: ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?



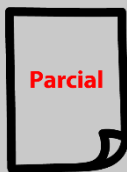
En la vida real, los enrutadores tienen una tabla de enrutamiento. Una tabla de enrutamiento guarda la distancia más corta para ir de un dispositivo a otro en la red. Un ejemplo de un enrutador es el ISR 4000 de Cisco. Otro ejemplo, es el que tiene en su casa para el Wifi

**3.6** Teniendo en cuenta lo anterior, para representar la tabla de enrutamiento, respondan: ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia?

**3.7** Calculen la complejidad de los ejercicios en línea, numerales 2.1 y [opcionalmente] 2.2, y agréguela al informe PDF

**3.8** Expliquen con sus palabras las variables (*qué es 'n', qué es 'm', etc.*) del cálculo de complejidad del numeral 3.7

#### 4) Simulacro de Parcial en el informe PDF

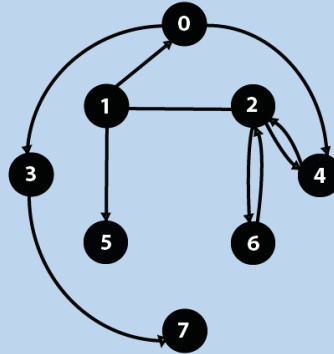


Para este simulacro, agreguen ***sus respuestas*** en el informe PDF.



***El día del Parcial no tendrán computador, JAVA o acceso a internet.***

1. Considere el siguiente grafo y complete la representación de **matrices de adyacencia**. Si no hay arco, por simplicidad, deje el espacio en blanco.



	0	1	2	3	4	5	6	7
0				1	1			
1								
2								
3								
4								
5								
6								
7								

2. Para el mismo grafo, completen la representación de **listas de adyacencia**. Como el grafo no tiene pesos, sólo se colocan los sucesores en la lista de adyacencia.

0 -> [3,4]

1 ->

2 ->

3 ->

4 ->

5 -&gt;

6 -&gt;

7 -&gt;

3. ¿Cuánta memoria (ojo, no tiempo sino memoria) ocupa una representación usando listas de adyacencia para el peor grafo dirigido con  $n$  vértices?

- a)  $O(n)$
- b)  $O(n^2)$
- c)  $O(1)$
- d)  $O(\log n)$
- e)  $O(n \cdot \log n)$

## 5. [Ejercicio Opcional] Lectura recomendada



"Quienes se preparan para el ejercicio de una profesión requieren la adquisición de competencias que necesariamente se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..."

Tomado de <http://bit.ly/2gJKzJD>



Véase Guía en **Sección 3, numeral 3.5 y 4.20** de la Guía Metodológica, "Lectura recomendada" y "Ejemplo para realización de actividades de las Lecturas Recomendadas", respectivamente

Posterior a la lectura del texto “**Robert Lafore, Data Structures and Algorithms in Java (2nd edition), Chapter 13: Graphs. 2002**” realicen las siguientes actividades que les permitirán sumar puntos adicionales:

- a) Escriban un resumen de la lectura que tenga una longitud de 100 a 150 palabras
- b) Hagan un mapa conceptual que destaque los principales elementos teóricos.



**NOTA 1:** Si desean otra lectura, consideren la siguiente: “**John Hopcroft et al., Estructuras de Datos y Algoritmos, Capítulo 6: Grafos dirigidos. Páginas 267 – 276. 1983**” que pueden encontrarla en biblioteca



**NOTA 2:** Estas respuestas también deben incluirlas en el informe PDF

## 5. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual



El trabajo en equipo es una exigencia actual del mercado. "Mientras algunos medios retratan la programación como un trabajo solitario, la realidad es que requiere de mucha comunicación y trabajo con otros. Si trabajas para una compañía, serás parte de un equipo de desarrollo y esperarán que te comuniques y trabajes bien con otras personas"

Tomado de <http://bit.ly/1B6hUDp>



Véase Guía en **Sección 3, numeral 3.6 y Sección 4, numerales 4.21, 4.22 y 4.23** de la Guía Metodológica

- a) Entreguen copia de todas las actas de reunión usando el tablero Kanban, con fecha, hora e integrantes que participaron

DOCENTE MAURICIO TORO BERMÚDEZ  
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627  
Correo: [mtorobe@eafit.edu.co](mailto:mtorobe@eafit.edu.co)

- b) Entreguen el reporte de *git*, *svn* o *mercurial* con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron
- c) Entreguen el reporte de cambios del informe de laboratorio que se genera *Google docs* o herramientas similares



**NOTA 1:** Estas respuestas también deben incluirlas en el informe PDF

## 6. [Ejercicio Opcional] Laboratorio en inglés:



El inglés es un idioma muy importante en la Ingeniería de Sistemas porque la mayoría de los avances en tecnología se publican en este idioma y la traducción, usualmente se demora un tiempo y es sólo un resumen de la información original.

Adicionalmente, dominar el inglés permite conseguir trabajos en el exterior que son muy bien remunerados

Tomado de [goo.gl/4s3LmZ](https://goo.gl/4s3LmZ)

Entreguen el código y el informe traducido al inglés. Utilicen la plantilla dispuesta en este idioma para el laboratorio

## Resumen de Ejercicios a Resolver

**1a** Realicen una implementación de la clase abstracta *Digraph*, llámela *DigraphAM* e implementen grafos con la estructura de datos Matrices de Adyacencia Etiquetadas

**1b** Creen la clase *DigraphAL* e implementen grafos con la estructura de datos Listas de Adyacencia.

**1.2** En la clase *GraphAlgorithms*, implementen un método que reciba como parámetro un grafo dirigido y que retorne cuál es el vértice que tiene más sucesores (vecinos).

**1.3** Prueben su código con los ejemplos contruidos en los numerales 1.1 y 1.2 para el *Algoritmo de Dijkstra*.

**1.4** Implementen un método que permita crear un grafo a partir de ese archivo del texto *medellin\_colombia-grande.txt*

**2.1** Resuelvan el ejercicio planteado

**2.3**[Ejercicio Opcional]: Resuelvan el siguiente problema <http://bit.ly/2gTLZ53>

**3.2**Incluyan una imagen de la respuesta de las pruebas del numeral 1.3

**3.2** Escriban una explicación entre 3 y 6 líneas de texto del código del numeral 1.1. Digan cómo funciona, cómo está implementado el grafo con matrices y con listas que hizo, destacando las estructuras de datos y algoritmos usados


**3.3** ¿En qué grafos es más conveniente utilizar la implementación con matrices de adyacencia y en qué casos en más convenientes listas de adyacencia? ¿Por qué?

**3.4** Para representar el mapa de la ciudad de Medellín del ejercicio del numeral 1.3, ¿qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?

**3.5** ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia? ¿Por qué?

**3.6** Para representar la tabla de enrutamiento, respondan: ¿Qué es mejor usar, Matrices de Adyacencia o Listas de Adyacencia?

**3.7** Calculen la complejidad de los ejercicios en línea, numerales 2.1 y [opcionalmente] 2.2, y agréguela al informe PDF

	<p>UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</p>	<p>Cód. ST0245 Estructuras de Datos 1</p>
---	---	---

**3.8** Expliquen con sus palabras las variables (*qué es 'n', qué es 'm', etc.*) del cálculo de complejidad del numeral 3.7

**4.** Simulacro parcial

**5. [Ejercicio Opcional]** Lectura recomendada

**6. [Ejercicio Opcional]** Trabajo en Equipo y Progreso Gradual

**7. [Ejercicio Opcional]** Laboratorio en inglés



## Ayudas para resolver los ejercicios

Ayudas para el Ejercicio 1.....	<a href="#"><u>Pág. 18</u></a>
Ayudas para el Ejercicio 1a.....	<a href="#"><u>Pág. 18</u></a>
Ayudas para el Ejercicio 1b.....	<a href="#"><u>Pág. 18</u></a>
Ayudas para el Ejercicio 1.3.....	<a href="#"><u>Pág. 18</u></a>
Ayudas para el Ejercicio 1.4.....	<a href="#"><u>Pág. 19</u></a>
Ayudas para el Ejercicio 2.1.....	<a href="#"><u>Pág. 21</u></a>
Ayudas para el Ejercicio 2.2.....	<a href="#"><u>Pág. 21</u></a>
Ayudas para el Ejercicio 3.3.....	<a href="#"><u>Pág. 21</u></a>
Ayudas para el Ejercicio 3.7.....	<a href="#"><u>Pág. 21</u></a>
Ayudas para el Ejercicio 4.....	<a href="#"><u>Pág. 22</u></a>
Ayudas para el Ejercicio 5a.....	<a href="#"><u>Pág. 22</u></a>
Ayudas para el Ejercicio 5b.....	<a href="#"><u>Pág. 22</u></a>
Ayudas para el Ejercicio 6a.....	<a href="#"><u>Pág. 22</u></a>
Ayudas para el Ejercicio 6b.....	<a href="#"><u>Pág. 22</u></a>
Ayudas para el Ejercicio 6c.....	<a href="#"><u>Pág. 23</u></a>

## Ayudas para el Ejercicio 1



**PISTA 1:** Si deciden hacer la documentación de los puntos del numeral 1 vean la *Guía en Sección 4, numeral 4.1 “Cómo escribir la documentación HTML de un código usando JavaDoc”*

## Ayudas para el Ejercicio 1a



**PISTA 1:** Un error común es retornar el peso de los arcos en lugar de los identificadores de los vértices en el método *getSuccessors*

## Ayudas para el Ejercicio 1b



**PISTA 1:** Un error común es retornar el peso de los arcos en lugar de los identificadores de los vértices en el método *getSuccessors*



**PISTA 2:** Véase *Guía en Sección 4, numeral 4.8 “Cómo definir una clase Pareja en Java”*

## Ayudas para el Ejercicio .1.3



**PISTA 1:** Véase *Guía en Sección 4, numeral 4.14 “Cómo hacer pruebas unitarias en BlueJ usando JUnit” y numeral 4.15 “Cómo compilar pruebas unitarias en Eclipse”*

## Ayudas para el Ejercicio 1.4



**PISTA 1:** Véase *Guía en Sección 4, numeral 4.13 “Cómo usar Scanner o BufferedReader”*



**PISTA 2:** Hay información que sobra, por ejemplo, la latitud y la longitud de cada vértice y el nombre de cada arista.



**PISTA 3:** Es mejor usar *BufferedReader* porque es más rápido que *Scanner*. La idea es leer en una cadena de caracteres el contenido de cada línea y usando el método *split* de la clase *String* o usando *StringTokenizer*, dividir la cadena en partes cada que hay una coma (,).



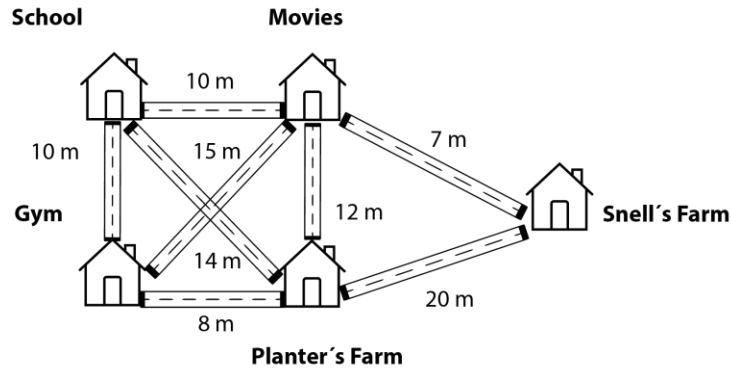
**PISTA 4:** Como los códigos no son secuenciales, es decir, no empiezan en cero y tampoco están todos los números consecutivos, una forma de manejar los vértices es usar un mapa (en Java, *HashMap* o *TreeMap*).

### Error Común





**Como un ejemplo**, para el siguiente mapa, el archivo de entrada es el siguiente:



**Vértices. Formato: ID, coordenada x, coordenada y, nombre**

```

10000 2.00000 0.00000 School
1 4.00000 1.00000 Movies
2 5.00000 2.00000 Snell
3 2.00000 5.00000 Planters
4 0.00000 2.00000 Gym
  
```

**Arcos. Formato: ID, ID, distancia, nombre**

```

10000 1 10.0 Calle 1
10000 3 14.0 desconocido
10000 4 10.0 desconocido
1 10000 10.0 Calle 2a
1 2 7.0 desconocido
1 3 12.0 desconocido
1 4 15.0 desconocido
2 1 7.0 desconocido
2 3 20.0 desconocido
3 10000 14.0 desconocido
3 1 12.0 desconocido
3 2 20.0 desconocido
3 4 8.0 desconocido
4 10000 10.0 desconocido
4 1 15.0 desconocido
4 3 8.0 desconocido
  
```

## Ayudas para el Ejercicio 2.1



**PISTA 1:** Usen un algoritmo para corroborar si es un grafo bipartito. Léase qué es un grafo bipartito en <http://bit.ly/2hGwAo2>



**PISTA 2:** Si desean, pueden usar DFS o BFS para resolver este problema, pero existe otro tipo de algoritmos para resolverlo también.



**PISTA 3:** Spoiler Alert! En este sitio web explican un algoritmo para verificar si un grafo es bipartito <http://bit.ly/2lOsQFZ>

## Ayudas para el Ejercicio 2.2



**PISTA 1:** Utilicen **Búsqueda en Profundidad** (Siglas en inglés DFS)



**PISTA 2:** Véase Guía en **Sección 4, numeral 4.13** “Cómo usar Scanner o *BufferedReader*”

## Ayudas para el Ejercicio 3.3



**PISTA 1:** <http://bit.ly/2qzZPLD>



**PISTA 2:** <http://bit.ly/2qSMq1Z>

## Ayudas para el Ejercicio 3.7



**PISTA 1:** Véase **Guía en Sección 4, numeral 4.11** “Cómo escribir la complejidad de un ejercicio en línea”

## Ayudas para el Ejercicio 4



**PISTA 1:** Véase *Guía en Sección 4, Numeral 4.18* “Respuestas del Quiz”



**PISTA 2:** Lean las diapositivas tituladas “*Data Structures II: Graph Implementation*”, encontrarán la mayoría de las respuestas

## Ayudas para el Ejercicio 5a



**PISTA 1:** En el siguiente enlace, unos consejos de cómo hacer un buen resumen <http://bit.ly/2knU3Pv>



**PISTA 2:** [Aquí](#) le explican cómo contar el número de palabras en Microsoft Word

## Ayudas para el Ejercicio 5b



**PISTA 1:** Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://www.mindmup.com/#m:new-a-1437527273469>

## Ayudas para el Ejercicio 6a



**PISTA 1:** Véase *Guía en Sección 4, Numeral 4.21* “Ejemplo de cómo hacer actas de trabajo en equipo usando Tablero Kanban”

## Ayudas para el Ejercicio 6b



**PISTA 1:** Véase *Guía en Sección 4, Numeral 4.23* “Cómo generar el historial de cambios en el código de un repositorio que está en svn”

## Ayudas para el Ejercicio 6c



**PISTA 1:** Véase Guía en Sección 4, Numeral 4.22 “**Cómo ver el historial de revisión de un archivo en Google Docs**”